

ИМПЛЕМЕНТАЦИЈА НОВИХ ТЕХНОЛОГИЈА У РЕАЛИЗАЦИЈИ НАСТАВЕ ИЗ ПРОГРАМИРАЊА

Milena Vesic¹ Milanko Kragovic² Nenad Kojic³ Luka Lukic⁴ Dimitrije Borcanin⁵

Резиме: Програмирање је доживело велике промене у последњих десетак година, поготово из области web програмирања. Промене су се одразиле у појави великог број библиотека и окружења, како код клијентских језика, тако и код серверских – од web технологија за организацију web страна, преко програмских језика за кориснички део апликација до програмских језика за манипулацију серверским делом апликације. Развојем технологија постоји потреба за што већим бројем људи који ће познавати те технологије и пратити будући развој. Поставља се питање како одлучити шта треба издвојити од свих новина и представити студентима у реализацији наставе. Постоји бојазан да уколико се нека област изостави, да ће студенти бити ускраћени за познавање неког новог оквира или библиотеке с обзиром на брзи развој технологија, док је с друге стране немогуће представити све библиотеке и оквири из области web програмирања због временског ограничења трајања наставе. У овом раду представиће се искуства и дилеме у реализацији наставе web програмирања на Високој ICT школи у Београду.

Кључне речи: настава програмирања, едукација, web програмирање, савремене технологије

IMPLEMENTATION OF NEW TECHNOLOGIES IN THE REALIZATION OF TEACHING IN PROGRAMMING

Abstract: Programming has undergone major changes in the last ten years, especially in the field of the web. The changes have been reflected in the emergence of a large number of libraries and frameworks, both in client side languages and server side languages – starting from web technologies for organizing web pages, through programming languages for the user part of applications to programming languages for manipulating the server part of the application. With the development of technologies, there is a need for as many people as possible who will know these technologies and follow future developments. The question is how to decide what should be separated from all the novelties and presented to students in the implementation of teaching. There is a fear that if an area is omitted, students will be denied knowledge of a new framework or library due to the rapid development of technology, while on the other hand it is impossible to present all libraries and frameworks in the field of web programming due to time constraints. This paper will present experiences and dilemmas in the implementation of web programming classes at the ICT College in Belgrade.

Key words: teaching programming, education, web programming, modern technologies

1. УВОД

Web програмирање је доживело енормну експанзију у последњих десетак година [1]. Сама та чињеница новоди да су се и web технологије и програмски језици развијали и проширивали кроз време. Те промене су приказане кроз многобројне верзије web технологија и програмских језика, кроз креирање њихових библиотека и оквира [1].

¹ Сарадник у настави, Академија техничко-уметничких струковних студија Београд одсек Висока школа за информационе и комуникационе технологије, Здравка Челара 16, Београд, milena.vesic@ict.edu.rs

² Предавач, Академија техничко-уметничких струковних студија Београд одсек Висока школа за информационе и комуникационе технологије, Здравка Челара 16, Београд, milanko.kragovic@ict.edu.rs

³ Професор, Академија техничко-уметничких струковних студија Београд одсек Висока школа за информационе и комуникационе технологије, Здравка Челара 16, Београд, nenad.kojic@ict.edu.rs

⁴ Сарадник у настави, Академија техничко-уметничких струковних студија Београд одсек Висока школа за информационе и комуникационе технологије, Здравка Челара 16, Београд, luka.lukic@ict.edu.rs

⁵ Сарадник у настави, Академија техничко-уметничких струковних студија Београд одсек Висока школа за информационе и комуникационе технологије, Здравка Челара 16, Београд, dimitrije.borcanin@ict.edu.rs

Уколико се направи ретроспектива и погледа се начин организације web страница, требало би споменути организацију страница кроз фрејмове, затим кроз табеле и на крају путем CSS-а, почевши од његове верзије 1, 2, 3 до верзије 4. И сам HTML као језик за креирање web страна је доживео експанзију и дошао до тренутно актуелних верзија 4 и 5 [2].

Поред организације web страна, CSS3 је омогућио да те странице оживе многобројим анимацијама, транзицијама и трансформацијама [3]. Такође, у односу на CSS2, омогућио је, оно што је данас незамисливо да не постоји, да web сајтови буду прилагодљиви за све врсте уређаја и екрана и донео нови начин организације самих страница [2]. Наравно, настали су и бројни оквири који помоћу својих класа омогућавају лакше креирање web страна које ће аутоматски бити прилагодљиве свим уређајима без додатног писања кода.

Статички сајтови су добили интерактивност имплементацијом JavaScript-а као клијентског језика [7]. Његова улога је била да омогући реакцију web сајта на корисникову акцију. Врло брзо потребе корисника расту, као и надметање у томе шта који језик има да понуди. JavaScript добија своју библиотеку jQuery коју данас користи преко 70% web сајтова [7]. Затим, настаје AJAX као сет техника у web окружењу за креирање асинхроних апликација, односно за слање података web серверу и преузимање истих. JSON се издвојио као JavaScript нотација за трансфер података од клијентског дела апликације до серверског дела апликације и обрнуто [8]. JavaScript је направио највећи преокрет када је као опште познат клијентски језик избацио оквир за подршку серверског кода, односно Node.js, и самим тим постао конкуренција серверским програмским језицима. Поред Node.js-а појавили су се бројне библиотеке и оквири, док су тренутно најактуелнији Angular, Vue и React.

За креирање динамичког сајта било је потребно познавање неког серверског програмског језика и рад са базом података. PHP као најкоришћенији серверски програмски језик за креирање динамичких web сајтова и web апликација, почео је од верзије 1, док је данас актуелна верзија 8. Доживео је промене у начину писања кода, од процедуралног, док није са верзијом 4 дошла и подршка за објектно оријентисано програмирање [10]. PHP је коришћен као подршка за серверски код за многе оквири и CMS (Content Management System) системе [10].

Све ове чињенице су довеле до неминовних потреба да се програмери непрекидно усавршавају и прилагођавају новонасталим стањима, као и образовни систем и начин обуке нових програмера који треба да познају и раније примењивану технологију, јер ће наследити у пракси велики број пројеката који су раније реализовани, али и новим пројектима где се очекују нове технологије. Ове промене представљају прави изазов за свакога ко или учи сам, или учи друге новим технологијама у овако турбулентним периодима промене програмских језика за потребе израде web презентација.

Циљ овог рада је да укаже на динамике промене и изазове који су пред образовним системом и организацији наставе у области програмирања сајтова. Рад представља анализу реалних ситуација, промена које су спровођене и питања која су остала отворена, а све у циљу да се нађе најбољи могући начин да се у оквиру ограниченог броја предмета и часова нађе најбољи начин да се ова материја презентује студентима у оквиру формалног образовања.

Рад ће бити представљен кроз анализу две области у домену интернет програмирања: статичке и интерактивне сајтове и биће организован кроз четири поглавља. У другом поглављу указаће се на искуства, решења и дилеме на које су аутори наилазили у реализацији наставе два предмета који се баве материјом статичких

сајтова. У трећем поглавау овај проблем анализираће се кроз други предмет којим се обрађује материја интерактивних сајтова и примене програмског језика JavaScript који захтева и претходно познавање основа програмирања и израде статичких сајтова. На крају, даће се закључак и списак коришћене литературе.

2. ИМПЛЕМЕНТАЦИЈА НОВИХ ТЕХНОЛОГИЈА У РЕАЛИЗАЦИЈИ НАСТАВЕ ИЗ ПРОГРАМИРАЊА

2.1. Web дизајн

Web дизајн представља почетни курс из области web програмирања на Високој ICT школи који се реализује у првој години. Као почетни курс нуди студентима основна знања из области web дизајна, односно омогућава студентима да науче како да креирају web сајтове. Испрва је било сасвим лако одабрати материју за предавања – HTML и CSS. HTML као језик за креирање web страна и CSS за стилизовање управо тих страница. Материја је и даље иста, али је с временом постала преобимна.

На Високој ICT школи настава се реализује у три триместра. Један триместар има десет недеља за реализацију предавања. Како би студенти што лакше усвојили материју, предавања започињу упознавањем са основама Интернета и web-а. Затим се студентима представљају основе HTML-а, и сви потребни тагови за организацију странице и убацивање садржаја, потом CSS својства за стилизовање, XML и SEO.

Како се временом показало, студентима је најтеже да савладају организацију страница. У тој материји долази до интеграције CSS-а са HTML-ом. Стога је промењен план да се након упознавања са таговима потребним за организацију, предаје CSS који ће омогућити стилизовање организације, па након тога HTML тагови за убацивање садржаја у виду табела и формулара.

На овај начин су студенти лакше усвајали поменути материју, међутим након неколико недеља учења нових HTML тагова за садржај, полако бледи знање стечено у претходном периоду. Поред поменуте проблематике у извођењу наставе, проблем настаје код рапидних промена које се дешавају у web технологијама.

Као што је у Уводу већ наведено, све web технологије су доживеле своју експанзију, самим тим и CSS спада у ту категорију. Нова верзија, тј. тренутно најактуелнија, верзија 3 настала је како би задовољила потребе корисника и тржишта.

С обзиром на то да се организација сајта реализује путем CSS-а, студентима почетком курса буде представљено својство *float* које омогућава да се блокови пакују један уз други. Међутим, CSS нуди и алтернативе својством *display* са вредностима *grid* и *flex* [2].

Табела 1 – Организација сајта путем CSS-а

Својство	Вредност
float	left / right
display	grid
display	flex

Flex box и *grid* омогућавају креирање web страница прилагодљивих за све врсте уређаја [3]. Постојање више могућности како би се добио исти резултат пружа web

дизајнерима прилику да изаберу оно што највише одговара њиховим афинитетима, док предавачима са ограниченим фондом часова прави проблем. Проблем се одражава у следећем: Како са ограниченим фондом часова изабрати који начин организације представити студентима? Данашњица намеће да web сајтови морају бити прилагодљиви свим врстама уређаја. Да ли представити својство *float* и ослонити се на CSS 3 правило *media query* за креирање прилагодљиве организације, или представити *flex box* који је скуп оба претходно наведена својства?

Уколико се студентима за кратак временски период само представе све могућности које нуди CSS постоји бојазан да се неће усвојити све на правилан начин и да ће им се информације само нагомилати. С друге стране, уколико се предавачи одреде само за један начин организације web страница, није у реду оставити студенте ускраћеним за познавање алтернатива које web технологије нуде. Поред тога што ће бити ускраћени за информације, може се створити дугорочан проблем који ће се манифестовати већ на следећем курсу у школи.

Тренутни план и програм предмета је такав да је главни фокус на организацији страница уз помоћ својства *float*, док се у недељи када се обрађује CSS3 показује *flex box* као алтернативно решење примењено од стране многих *front-end* оквира.

2.2. Практикум из web дизајна

Овај предмет се реализује у триместру након одслушаног базног курса из Web дизајна. Курс Практикум из web дизајна обухвата области које се односе на *Responsive Web Design* и напредни CSS. Један од CSS оквира који се предаје на овом курсу је *Bootstrap*. *Bootstrap* представља CSS оквир који омогућава креирање сајта који је прилагодљив свим врстама уређаја [4]. Класе за организацију страница овог оквира су базиране на својству *display* са вредношћу *flex*. *Foundation* је такође оквир који је базиран на *flex*-у [5]. Међутим, *Materialize* представља оквир чије су класе за организације базиране на својству *float* [6]. Зато се морало добро извагати шта предавати и представити студентима у претходном курсу.

С обзиром на фонд часова за овај предмет немогуће је приказати све CSS оквири, стога су изабрани они који су најзаступљенији и који се највише користе на тржишту. Поред *Bootstrap-a* и *Materialize-a* на овом предмету су своје место добили и скриптни језици CSS-a, *Sass* и *Less*, као и детаљнија објашњења за писање правила за *responsive web design* уз помоћ CSS3 *media query*-ја.

У првој години извођења овог предмета, материја је презентована студентима редоследом који је приказан у претходном пасусу. Студенти су лепо усвајали оквири који нуде већ готове, дефинисане класе које им омогућавају да креирају странице које су прилагодљиве свим врстама уређаја. Враћање на CSS3 и *media query* је стварало одбојност јер су већ били упознати са оквирима који имају већ написан CSS код, који су они сада морали да испишу ако не користе неки оквир.

Стога се у наредним годинама, у првим недељама триместра обрађивало правило *media query* и тек када савладају самостално писање *responsive web design-a*, престављају им се оквири. Овај начин је био прихватљивији за студенте, међутим, проблем је представљало савладати два оквира. Студенти су већински успевали да половично савладају оба оквира. Инцијално, ова два оквира су изабрана због њихове заступљености приликом израде web сајтова и web апликација, али због немогућности усвајања знања о њима више од 80%, нови приступ је да се у другој половини триместра предаје само један оквир, док ће се сви остали само поменути. Поред

немогућности усвајања знања о оба оквира, проблем је представило и ново издање оквира *Bootstrap* које је било проширено великим бројем класа.

Након повратних информација добијених од стране студената, а и самим ограничењем броја часова доступних за овај предмет, план и програм је реализован кроз предавања о *Responsive web design*-у и примени *media query*-ја за креирање web страница прилагодљивих свим врстама уређаја у првим недељама триместра, док се у другом делу триместра поред скрипних језика CSS-а обрађује оквир *Bootstrap*.

3. WEB ПРОГРАМИРАЊЕ

Након курсева помоћу којих су студенти у стању да самостално направе дизајн сајта прилагодљив свим врстама уређаја, или коришћењем неког оквира, следећи логичан корак је предмет у коме се изучава неки клијентски језик. Дуго низ година избор представља *JavaScript*.

Као опште познат клијентски језик који је унео интерактивност у статичке сајтове, за његово презентовање био је потребан један предмет. Овај предмет је студентима представљен у другој години студија. Студентима су се представљале основе језика, методе, функције, итд., као и његова имплементација у web сајтове. Поред основа, представљена је библиотека *jQuery* и *AJAX* за размену података са web сервером. *AJAX* се користи за асинхрону комуникацију, односно за размену информација клијентског и серверског дела апликације, и у ранијим годинама, у *XML* формату [10].

На основу вишегодишњег искуства, може се истаћи да је овај део материје био најпроблематичнији за студенте. Код којим се омогућавао овај пренос података је за студенте био непојмив, стога је број недеља које су се користиле за предавање ове материје често био мењан.

Напредовањем технологија и избацивањем нових верзија стандарда и библиотека, постало је немогуће све основе и новине представити у оквиру једног предмета. *jQuery* је добијао нове функционалности, самим тим и нове верзије, *JavaScript* је избацио *JSON* као своју посебну нотацију за пренос података која је нашла велику примену у реалним, „живим” апликацијама, па је и то требало убацивати у предавања [9]. Новом акредитацијом је дошло до поделе овог предмета на два.

Подела предмета довела је до растеређења и лакшег усвајања информација. Иако се у првом делу предмета обрађују основе, с временом је и то постало обимно због великих промена које се дешавају на годишњем нивоу. Из тог разлога је потребно стално праћење званичних сајтова са документацијом програмског језика, јер се дешава да припремљени материјал можда садржи неку методу која је по новој верзији стандарда застарела. Студентима је немогуће приказати и показати примену сваке методе и функционалности коју нуди тај програмски језик због временског ограничења триместра и броја часова који су намењени за тај предмет.

Други предмет обухвата градиво о објектима јер су сви оквири и библиотеке произашле из овог програмског језика објектно оријентисане. С друге стране приказује се комуникација путем *AJAX*-а са серверским делом апликације, што пре десет до петнаест година није био случај. Знала се тачна подела који су клијентски, а који серверски језици и њихова интеграција није била примењивана у великој мери.

Данас је немогуће замислити апликацију у којој не постоји и клијентски и серверски део кода, стога су се предмети који су били засебни и изучавали клијентске и серверске језике, новом акредитацијом у нашој Школи поделили на два нова предмета.

Што се тиче градива, студенти су били растерећени, али се број предмета повећавао. Посматрајући динамичке web апликације, није било довољно само показати основе програмског језика и његову интеграцију са базом података. Како *PHP* омогућава и процедурално писање кода и објектно оријентисан приступ, немогуће је одлучити се за један начин и њега представити студентима [10]. Након усвајања једног начина писања кода, студентима увек буде проблем да се навикну на други начин писања. Како би били спремни за тржиште, немогуће је избацити неки од описаних начина. Објектно оријентисан приступ, поред тога што је заступљен од стране многих програмских језика, представља принцип на коме су засновани многи оквири *PHP* програмског језика [10].

Зато је било добро решење да се у једном предмету примењује процедурални начин писања кода, док се у другом предмету примењује објектно оријентисани приступ, а самим тим се студенти припремају за предмете треће године, где се изучавају предмети који представљају оквире и клијентских и серверских језика, базирани на објектно оријентисаном принципу.

Избор оквира је представљао велику дилему. Како изабрати оквир, а не ускратити студентима могућност да се након завршеног школовања пријаве на неки конкурс за посао јер ће се тражити познавање оквира који нису изучавали у току школовања. За избор оквира су пресудили конкурси за посао, односно потребе тржишта.

Како би студенти имали могућност да усвоје сво знање из области web програмирања где се обрађује *JavaScript* као клијентски језик, у последњој акредитацији креирана су два предмета, *Web програмирање 1* и *Web програмирање 2*. За обраду серверског језика *PHP* и његову интеграцију са *JavaScript-ом* такође су креирана два предмета, *Web програмирање PHP* и *Практикум из Web програмирања PHP*. Сви наведени предмети се слушају у другој години студија.

Након ових предмета студенти су спремни за предмете који се реализују у трећој години студији, а обрађују оквире креиране на основу поменутог клијентског и серверског језика.

4. ЗАКЉУЧАК

Уз десетогодишње искуство може се извести само један закључак, немогуће је представити све програмске језике, оквире и библиотеке. Трудимо се да будемо у складу са тржиштем и са новинама које се појављују сваког дана. Редовно истраживање тржишта нам омогућава да се одредимо које ћемо програмске језике представити, као и како одабрати оквире и библиотеке.

Студентима константно скрећемо пажњу на рапидну експанзију у области web програмирања. Поред области које им представљамо, увек напомињемо шта још додатно постоји и које су алтернативе, и упућујемо их на стално истраживање и проширивање знања које поседују. Тудимо се да пратимо њихове реакције на предавањима и усклађујемо план и програм како би они најлакше усвојили знање које им преносимо.

Због сталног надограђивања постојећих верзија програмских језика, оквира и библиотека, као и због честог појављивања нових оквира и библиотека, било би

потребно креирати додатне предмете који покривају ову област или променити, тј. повећати фонд часова на постојећим предметима како би студенти имали времена да прихвате презентоване материјале, а и како би предавачи успели да пренесу што више знања из поменутог области. Додатно решење би било и креирање новог модула у оквиру смера Интернет технологије који ће имати више предмета који се баве облашћу Web програмирање.

На овај начин би предавачима било лакше јер не би морали процењивати шта је најмање штетно по студенте да се избаци из материје, док би студенти имали могућност да више материје науче у формалном образовању него да се ослањају на самостално учење, а самим тим били би и спремнији за тржиште и проналажење посла.

5. ЛИТЕРАТУРА

- [1] Chris Aquino, Todd Gandee. (2016). *Front-End Web Development*. Big Nerd Ranch Guides.
- [2] Ben Frain. (2020). *Responsive Web Design with HTML5 and CSS*. Packt Publishing.
- [3] John Rhea. (2020). *Humanity's Second Chance: Interactive HTML, Advanced CSS and Responsive Design*. Undead Institute.
- [4] Jörg Krause. (2020). *Introducing Bootstrap 4*. Apress.
- [5] *What is flexbox?*. (2020). Преузето 15. новембра 2020, са <https://get.foundation/learn/what-is-flexbox.html>
- [6] *Grid*. (2020). Преузето 15. новембра 2020, са <https://materializecss.com/grid.html>
- [7] Paul McFedries. (2018). *Web Coding & Development All-in-One*. For Dummies.
- [8] Matt Frisbie. (2019). *Professional JavaScript for Web Developers*. Wrox.
- [9] John Bach. (2020). *JavaScript and JQuery*. mEm Inc.
- [10] Robin Nixon. (2018). *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. O'Reilly Media.